

UNITED STATES PATENT APPLICATION

FOR

DISK FRAGMENTATION TEST SYSTEM

Inventor(s):

Michael Joseph JOHNSON

Sawyer Law Group LLP
2465 E. Bayshore Road, Suite 406
Palo Alto, California 94303

I hereby certify that this paper and/or fee is being deposited with the United States Postal Service "EXPRESS MAIL POST OFFICE TO ADDRESSEE" service under 37 CFR 1.10 on the date indicated below and is addressed to: Assistant Commissioner for Patents, ~~Washington, D.C. 20231~~ **PO Box 1450, Alexandria, VA 22313-1450**

Michael Johnson
Signature

DATE OF DEPOSIT: July 30, 2003

EXPRESS MAIL LABEL NO. E1331728295uS

Inventors: Michael J. JOHNSON

DISK FRAGMENTATION TEST SYSTEM

FIELD OF THE INVENTION

The present invention relates generally to fragmenting a storage device of computing system, and more specifically to fragmenting a hard disk used in a computing system under test.

5 BACKGROUND OF THE INVENTION

The tendency of files recorded onto a hard drive of a computing system to become fragmented over time is well known. It is recognized, qualitatively, that hard drive fragmentation reduces a speed of certain disk input/output (I/O) operations. Users of many types of computing systems are instructed to periodically use disk defragmentation tools to help
10 address performance degradation arising from fragmented files written to and accessed from the hard drive. Since these tools are not available on all platforms, and the frequency and/or timeliness of application of these tools is not assured in many environments, it is likely that some degree of fragmentation exists on most consumer systems.

However, in planning machine requirements or server capacity, particularly for new
15 system designs, allowance is made for some degradation of disk I/O performance over time.

The magnitude of the performance degradation related to varying degrees of fragmentation is not at all known, and it is very difficult to determine, *a priori*, what specific effect will be manifested for a specific degree of fragmentation. In some cases, a degree of file fragmentation that causes little degradation can cause extreme degradation in another environment.

5 Part of the difficulty in evaluating the relationship between disk I/O performance and file fragmentation is that a response time of any given software or hardware function may depend upon a wide variety of factors, not just disk I/O performance. For example, network turnaround, physical device wait conditions, multithread locks, or human operator interaction may dominate the performance of a particular function.

10 US Patent No. 5,778,392 describes an algorithm that scatters files already existing on a disk across a disk in order to fragment it. US Patent 5,857,101 describes a method for breaking files into fragments in order to store them in page-accessed order. US Patent 5,727,185 includes a method for subdividing available storage into two or more categories with the ability to shift free space between the categories.

15 Accordingly, what is needed is a system and method for intentionally producing varying degrees of hard file fragmentation on a hard drive of a test system to assist in the evaluation and measurement of disk I/O performance as a function of file fragmentation. The present invention addresses such a need.

SUMMARY OF THE INVENTION

20 Disclosed is a method for preparing a disk drive of a computing system for test, including the steps of recording a plurality of test files to a partition of the disk drive until a free space of the partition is less than a predetermined value; and thereafter deleting a portion of the

plurality of test files from the partition. A computer program product including a computer readable medium carrying a program for preparing a disk drive of a computing system for test, the computer program product having code for recording a plurality of test files to a partition of the disk drive until a free space of the partition is less than a predetermined value; and code for deleting a portion of the plurality of test files from the partition.

By selectably fragmenting files of a hard drive used in a test computing system, performance effects of progressively worse fragmentation may be easily evaluated and measured, allowing a system designer to potentially reduce, minimize or eliminate some or all of the negative effects of fragmentation.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a schematic block diagram of a preferred embodiment for a test computing system including a hard drive to be fragmented; and

Figure 2 is schematic block diagram of a preferred embodiment for a hard disk fragmentation control process.

DETAILED DESCRIPTION

The present invention relates to purposeful fragmentation of files of a hard drive in a test system used to evaluate and measure disk I/O performance as a function of degrees of fragmentation. The following description is presented to enable one of ordinary skill in the art to make and use the invention and is provided in the context of a patent application and its requirements. Various modifications to the preferred embodiment and the generic principles and features described herein will be readily apparent to those skilled in the art. Thus, the present invention is not intended to be limited to the embodiment shown but is to be accorded

the widest scope consistent with the principles and features described herein.

Figure 1 is a schematic block diagram of a preferred embodiment for a test computing system 100 including a hard drive 105 to be fragmented. Computing system 100 further includes a central processing unit (CPU) 110 coupled to a read-only memory (ROM) 115 for controlling hard drive 105 through a host adapter 120 communicating with an integrated drive electronics (IDE) controller manufactured into hard drive 105.

Hard drive 105 typically provides its storage volume divided into one or more logical partitions that may be treated as independent drives by computing system 100. Hard drive 105 has a primary partition that may be designated as being bootable, meaning it includes a boot sector 125 including instructions for initializing the desired operating system. Hard drive 105 often may also include files for an operating system (OS) 130, one or more program(s) 135 and/or data 140. Computer system 100 further includes an input/output (I/O) system 150 that incorporates necessary controllers for transferring data between CPU 110 and other components. These components may include a removable media reader 155 (e.g., floppy diskette drive or optical disk drive – reader and/or writer), a monitor 160, a keyboard 165, a pointing device 170 (mouse, trackball, pen/stylus), an imaging system 175 (scanner, camera, printer), a communications system 180 (network, modem, wireless) and/or an audio system 185. Computing system 100 may include one or more of some of the components depending upon the intended uses and implementations. A computer readable medium 190 (e.g., a floppy disk) may store instructions and program code elements in a non-volatile format that are executable by CPU 110 to implement an embodiment of the present invention. These elements may be accessed through one of the components, or stored, in some alternate preferred embodiments, on one or more partitions of hard drive 105. The preferred embodiment is

described in terms of fragmenting files on hard disk 105 for ease in description. It is understood that the term hard disk includes other types of storage devices that record file information in similar fashion, and includes sub-divisions of such devices (e.g., logical partitions treated as independent hard drives by the OS).

5 In operation, when computer system 100 is started, basic input-output settings (BIOS) stored in ROM 115 activate basic I/O devices (e.g., hard disk 105, CPU 110, media reader 155, monitor 160, keyboard 165, and mouse 170). As CPU 110 gets activated, it is instructed to access one or more memory locations to begin the process of starting the OS. Typically, CPU 110 is directed to boot sector 125 of hard drive 105 to initialize OS 130. Boot instructions
10 and/or OS 130 may be provided in other non-volatile memory locations of computer system 100, including medium 190.

OS 130 controls the overall operation of computing system 100 and activates all the I/O systems and communicates between a user and computing system 100 and the various subsystems. Programs and data used in computing system 100 are recorded on hard disk 105
15 and accessed in response to OS 130 and/or user interactions with computing system 100. The files on hard disk 105 are recorded in addressable units established on the recordable medium. It is rare for the size of any particular file and the size of the addressable unit to exactly match. When the file size is smaller, the unit is partially filled. When the file size is larger, multiple units are used to store the entire file contents. Computing system 100 keeps track of which
20 units, and in which order, define each stored file.

Preferably, all the units of a file are recorded in contiguous physical locations so the medium reading mechanism does not need to reposition itself to read the units in the proper order. Files written onto a newly formatted hard disk are often written in such fashion, with the

partition free space filled in a particular order as established in the configuration of computing system 100. However, once a file is deleted, freeing a certain number of units, it is purely coincidental whether a new file to be written uses exactly the same number of units. If a file needing fewer units is written, a “space” is left. Eventually a file will be written that does not fit
5 into the space, so the file is split into two or more pieces and written into non-contiguous spaces. During ordinary use, many files are written and deleted, resulting in more and more files being recorded into non-contiguous spaces. This characteristic of files being recorded in non-contiguous spaces is referred to as fragmentation. The present invention purposefully fragments files of a partition of hard disk 105 to enable the evaluation and measurement of the
10 performance of hard disk 105 and/or computing system 100 with hard disk 105 in varying degrees of fragmentation.

Figure 2 is schematic block diagram of a preferred embodiment for a hard disk fragmentation control process 200. Process 200 begins with an initialization step 205 that sets:
15 (i) a number of desired phases “P”, (ii) a file index “N” to zero, (iii) a file count “M” to zero, and (iv) a repetition number “R” for file write cycle retries. In the preferred embodiment, P is three and R is ten, though other values may be used in different implementations.

After initialization step 205, process 200 generates a random number S at step 210. S is preferably uniformly distributed between a MIN file size and a MAX file size, with MIN set equal to one byte and MAX set equal to fifty Megabytes for the preferred embodiment.

20 Process 200 next, step 215, tests whether a current freespace on the hard disk is less than S. If the freespace is not less than S, process 200 advances to step 220 from step 215, otherwise process 200 advances to step 225 when the freespace is less than S.

Step 220 increments N, creates file N, writes S bytes to newly created file N, and

increments M. Thereafter, process 200 returns to step 210. Step 210 through step 220 constitute the file write cycle of a phase of process 200. The test at step 225 determines whether the file write cycle tried R times to generate S less than the freespace. If not, process 225 returns to step 210 to begin another file write cycle. When the test at step 225 is positive and R retries have been made, the write cycle of a phase of process 200 is complete and process 200 begins a file delete cycle at step 230.

Step 230 sets M' equal to M. Next, step 235 generates a random number "F" between one and N. Thereafter, process 200 deletes file F (if it exists) and decrements M (if file F existed) at step 240. Process next, step 245, tests whether M'/2 files have been deleted. If not, process 200 returns to step 235 and if it has, process 200 concludes the file delete cycle and advances to step 250.

Step 250 tests whether the P number of phases have been completed. If not, process 200 returns to step 210 to begin another file write cycle and if they have, process 200 stops at step 255.

With the values set as specified above, process 200 fragments the files on a hard disk to realistically duplicate the condition of the files of the hard disk had the hard drive been used over the course of months and/or years in typical types of environments. This realistic fragmentation is achieved in a few phases, with the free space fragmented after the first phase, and the files heavily fragmented after successive phases. After each phase, the hard disk is roughly one-half full.

For the preferred embodiment, P is three and R is ten, though other implementations may use a different number of phases and write cycle repetitions. The purpose of the R repetitions is to inhibit process 200 from leaving a large block of unwritten freespace at the

completion of the write cycle, while helping to ensure that files generated near the end of the write cycles still have approximately the same statistical characteristics as the other generated files. Setting R equal to ten provides ten chances to pick random sizes S small enough to fit in the freespace available at the beginning of each phase. File sizes generated in any subsequent phase have a roughly ten percent chance of fitting into the left at the end of the current phase. In other words, ninety percent of the files generated in the subsequent phase will be too large to fit into the region(s) left unfragmented by the current phase (a desired result). Assuming hard disk size is large compared to MAX, the approximately nine “extra” chances at selecting a suitable size S will not severely affect the average file size, considering the average over the whole disk. However, if R were very large (e.g., a million), then the hard disk would end up with thousands of tiny files that would fill the free space completely but would severely skew the average file size, making the result unrealistic as a representation of extended hard drive usage. In some implementations it may be desirable to achieve such a result however, but it is not the desired outcome of the preferred embodiment.

The present invention for a monitoring service may be provided as a computer program product that may include a machine-readable medium having stored thereon instructions that can be used to program a computer (or other electronic devices) to perform a process according to the present invention. The machine-readable medium may include, but is not limited to, floppy disks, optical disks, CD-ROMS, magneto-optical disks, ROMS, RAMS, EPROMS, EEPROMS, magnetic or optical cards, or any type of media/machine-readable medium suitable for storing electronic instructions.

Although the present invention has been described in accordance with the embodiments shown, one of ordinary skill in the art will readily recognize that there could be variations to the

embodiments and those variations would be within the spirit and scope of the present invention.

Accordingly, many modifications may be made by one of ordinary skill in the art without departing from the spirit and scope of the appended claims.